MetroTrack

Predictive Tracking of Mobile Events using Mobile Phones

13.03.2012

Tobias Weber

Contributions

- Proof-of-concept implementation and evaluation of the first mobile(!) sensing system
- Use of off-the-shelf mobile phones
  - Already in use by possible participants
  - Cheaper to manufacture than specialized systems
- Predictive recovery protocol
- Improved tracking under varying sensor density

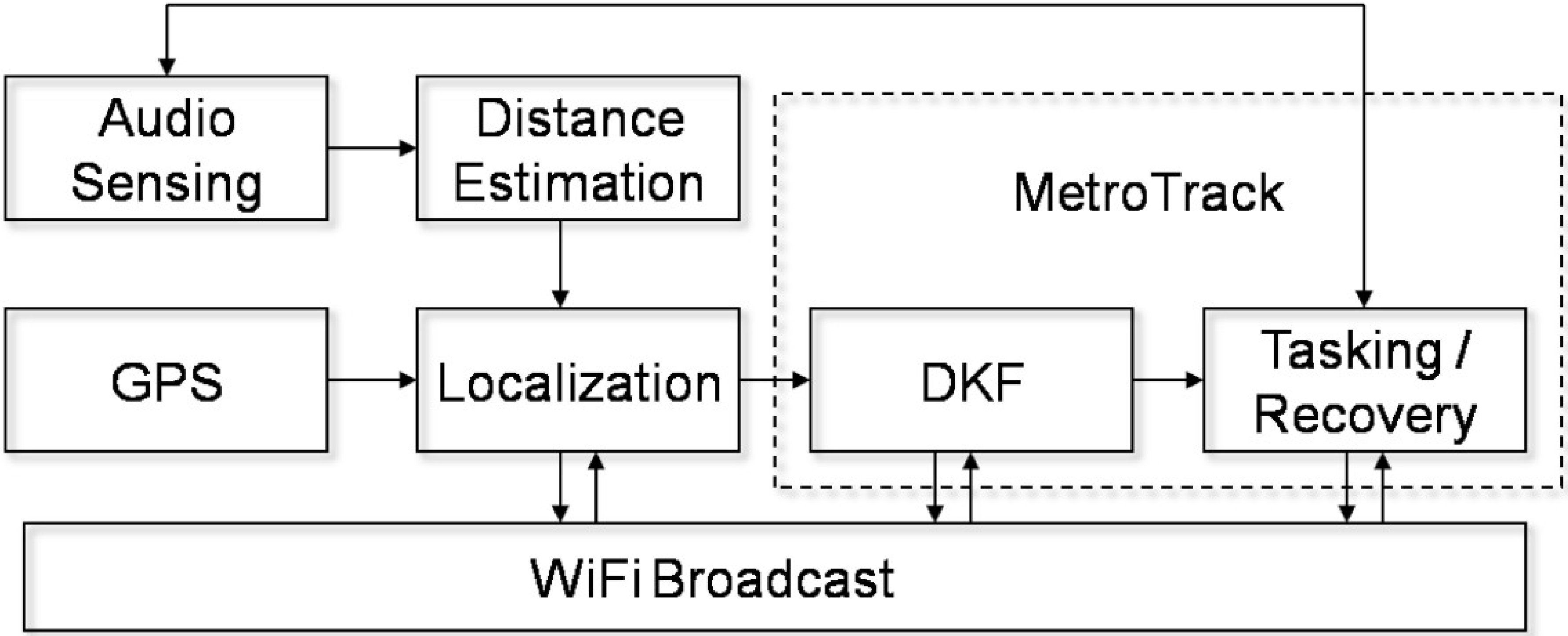
Goals of MetroTrack

- Many people now carry smartphones that can be used to build an opportunistic sensing network
    - When to start sensing?
- Disadvantages of static sensor networks
    - Area predetermined and limited
    - Wrong positioning very costly
- Temporally and spatially varying sensor density

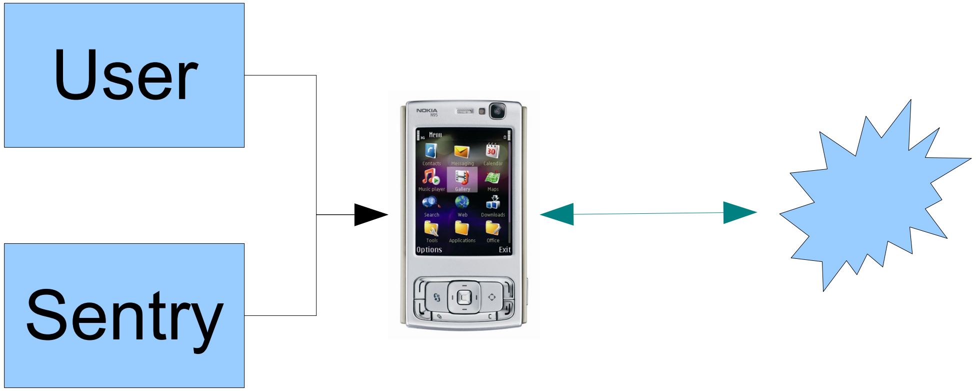
Understanding human behavior and routines

- Understanding human behavior is basis for sensor availability in MetroTrack application
    - Predict sensor density
- Investigating the impact of environmental events on our behavior
    - Pollution
    - Noise
- Especially in regard to mobility

MetroTrack architecture



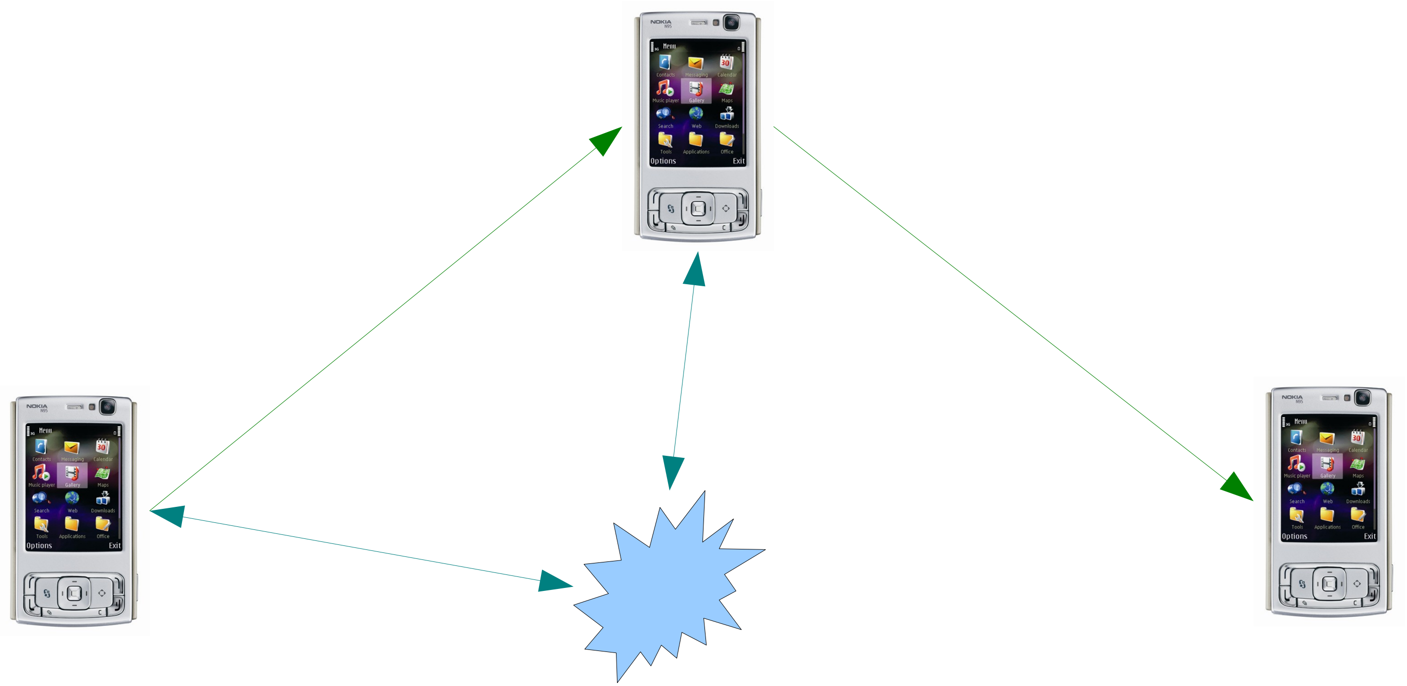
Tasking: Initiation



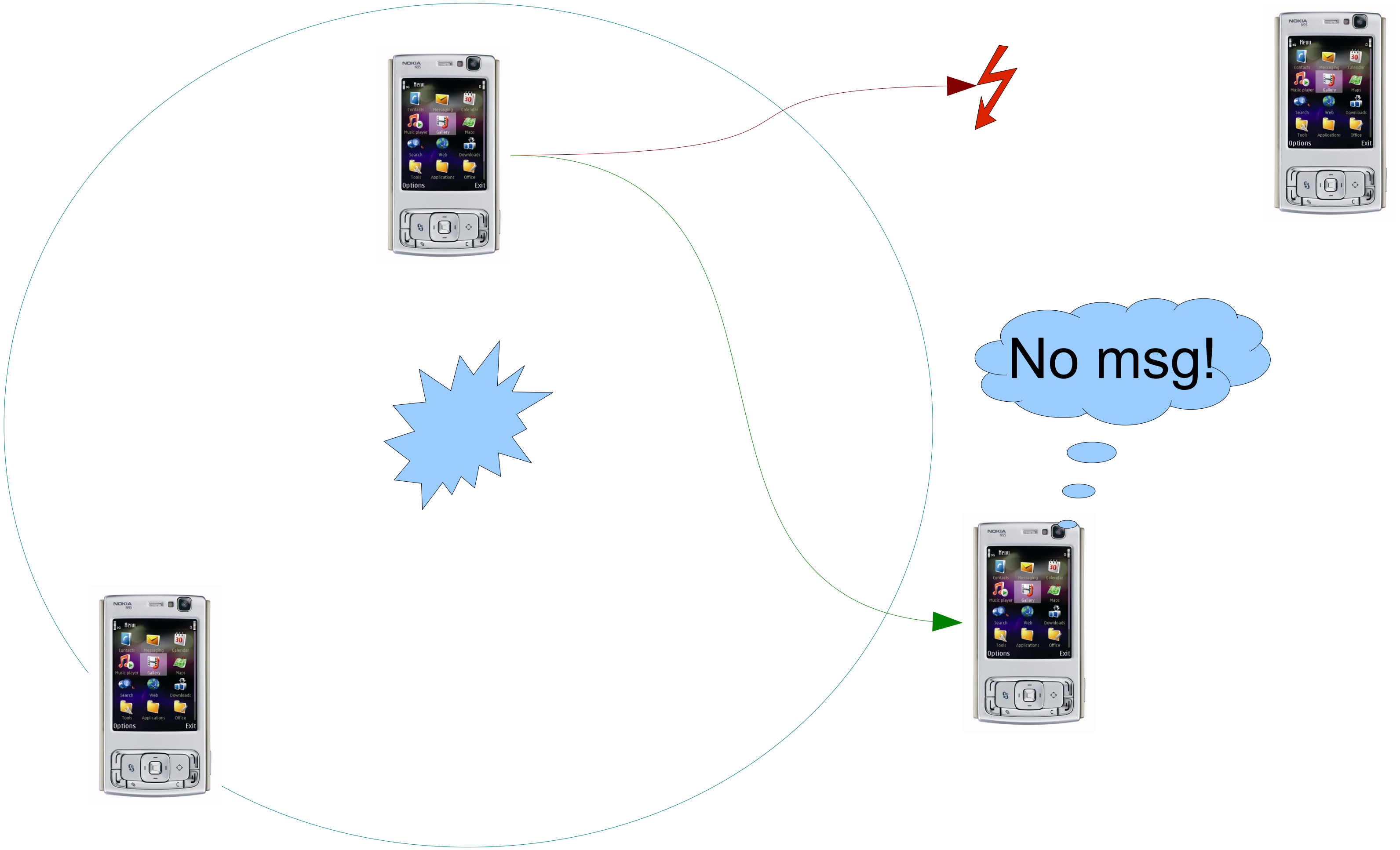
User

Sentry

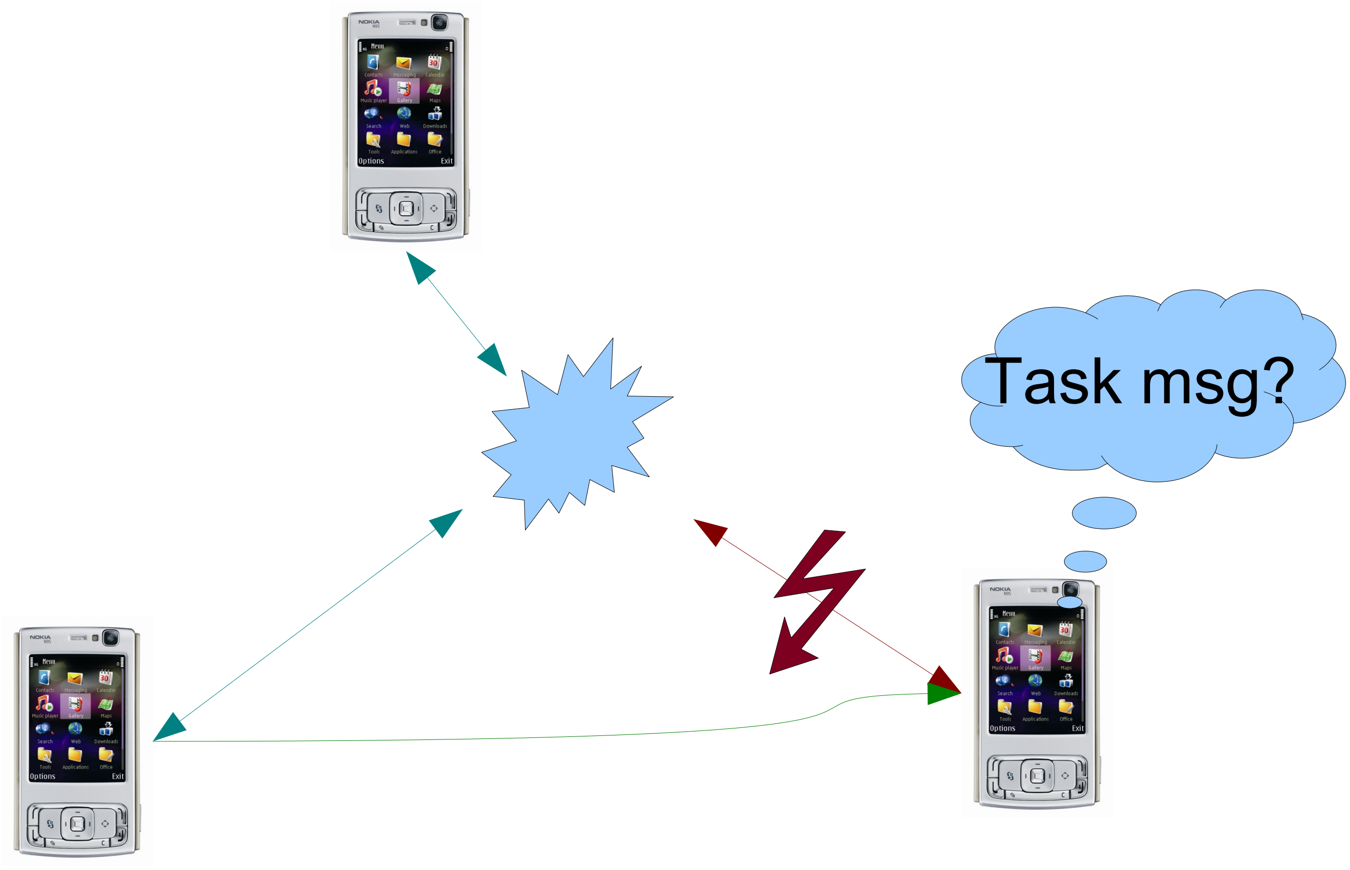
Message forwarding



Tasking region

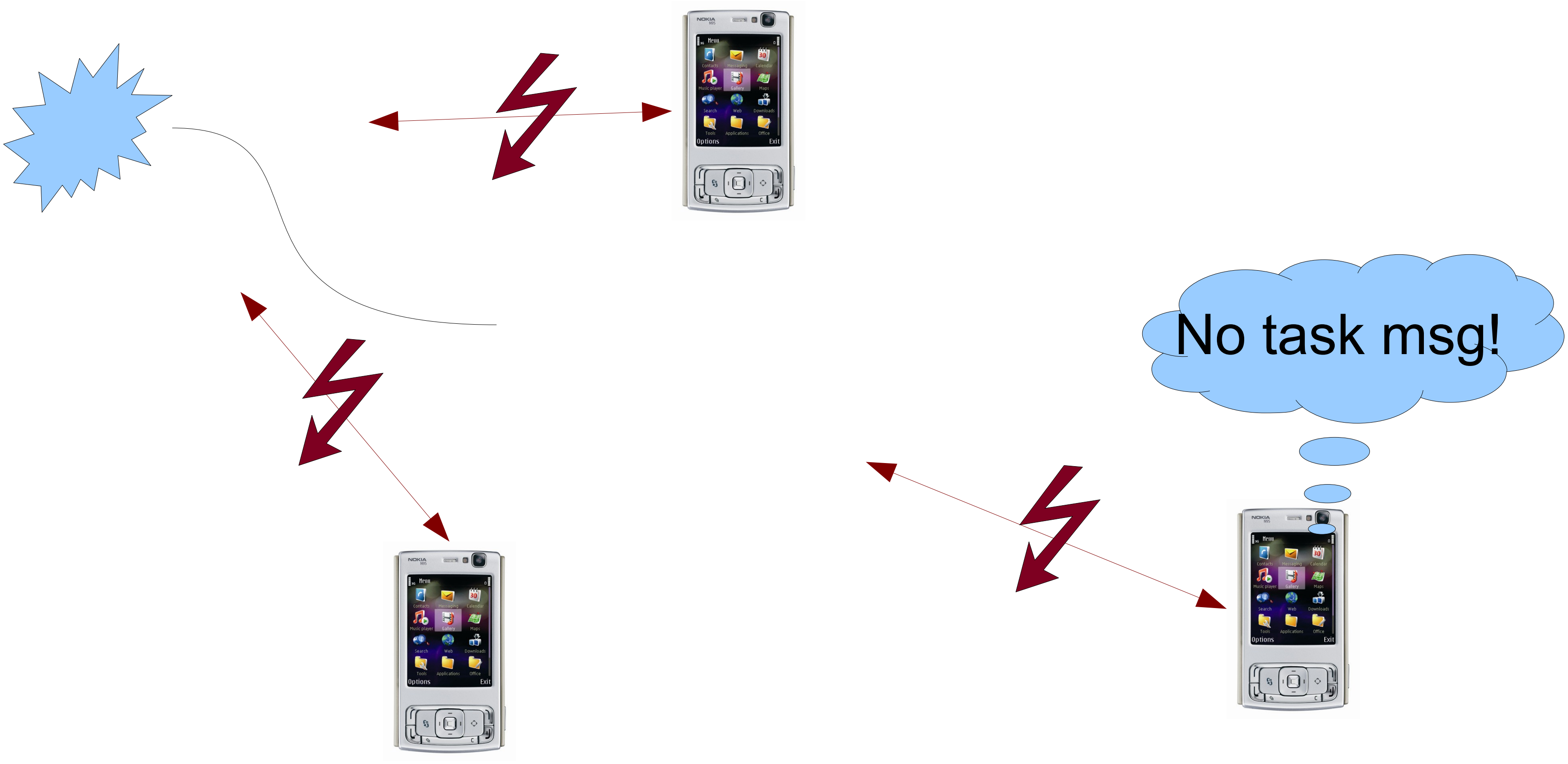


Recovery

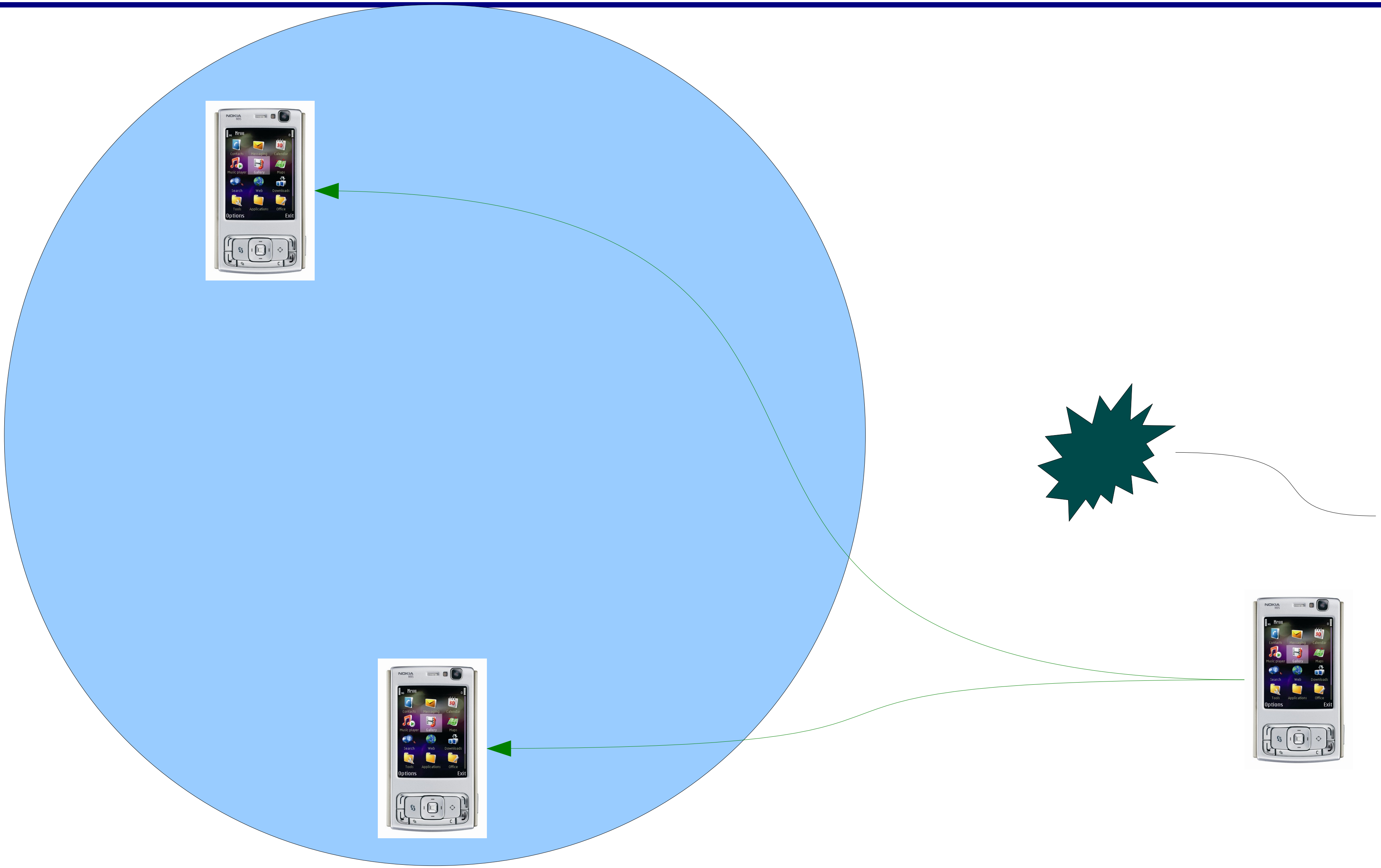


Task msg?

Target lost



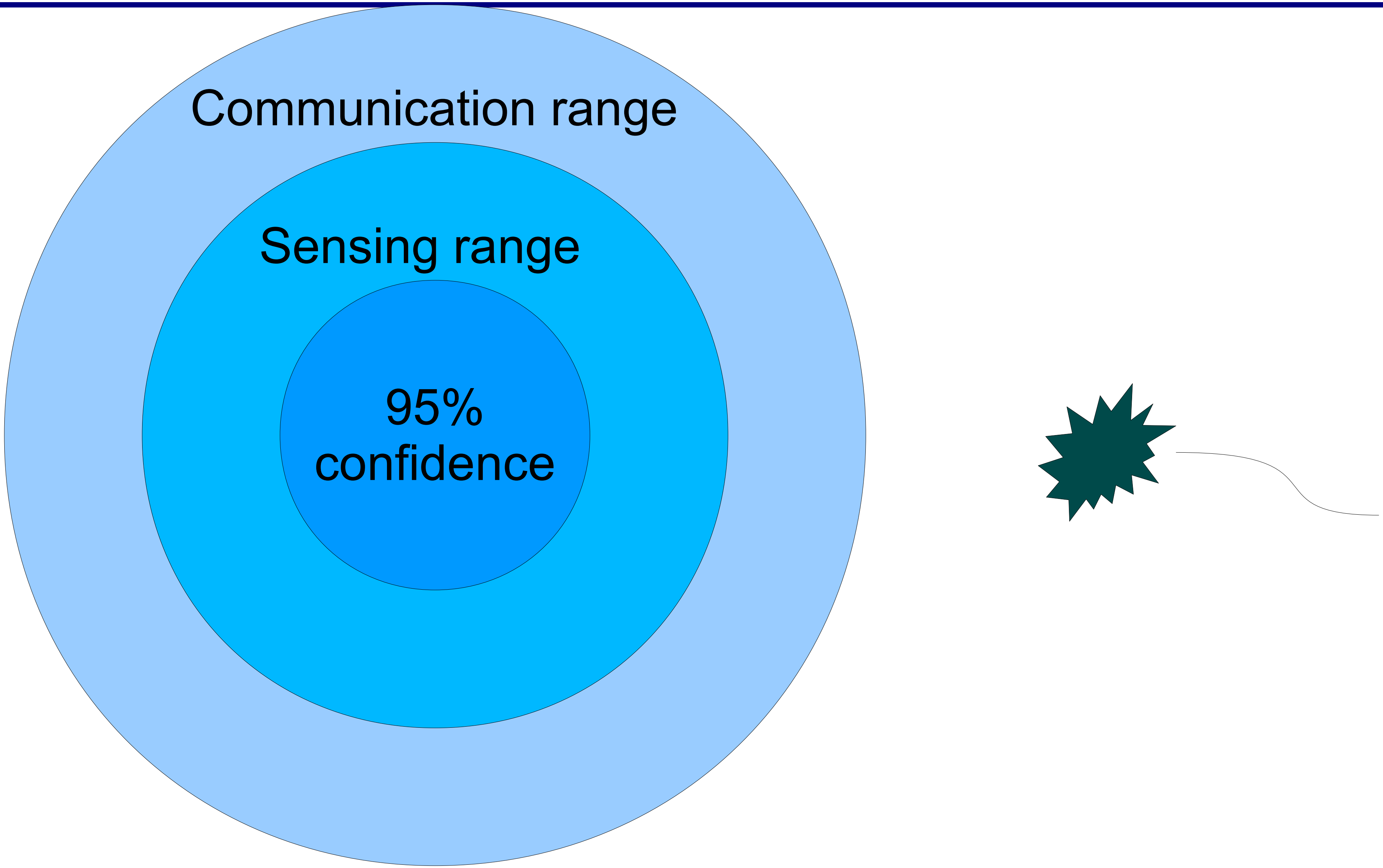
Recovery messages



Kalman filter

- Takes a vector containing last known speed and position
- Calculates new position assuming speed stays constant
- Adds a random deviation to speed and position
- Next step:
  - Starts with the deviated speed and position vector
  - Calculates new position and again adds deviation
- Random deviations have a Gaussian distribution
  - Confidence area for 95% has radius 2 times the deviation

Recovery area



Recovery end

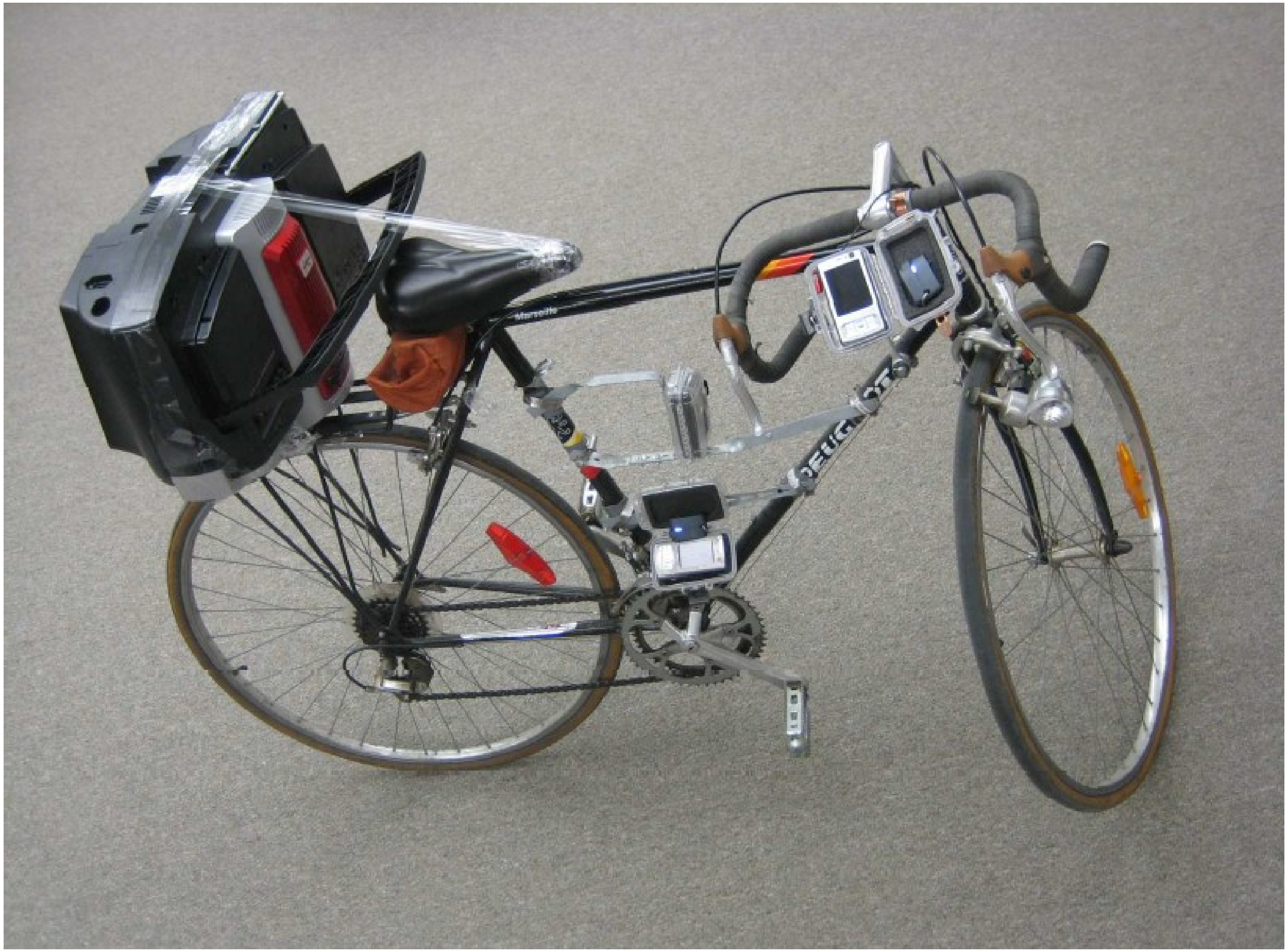
- Target has been detected
    - Tasking messages broadcasted by detector
- Tasking message has been received
- Sensor moves outside the recovery area
- Recovery timer expires
    - Tracking stops!

Experiment

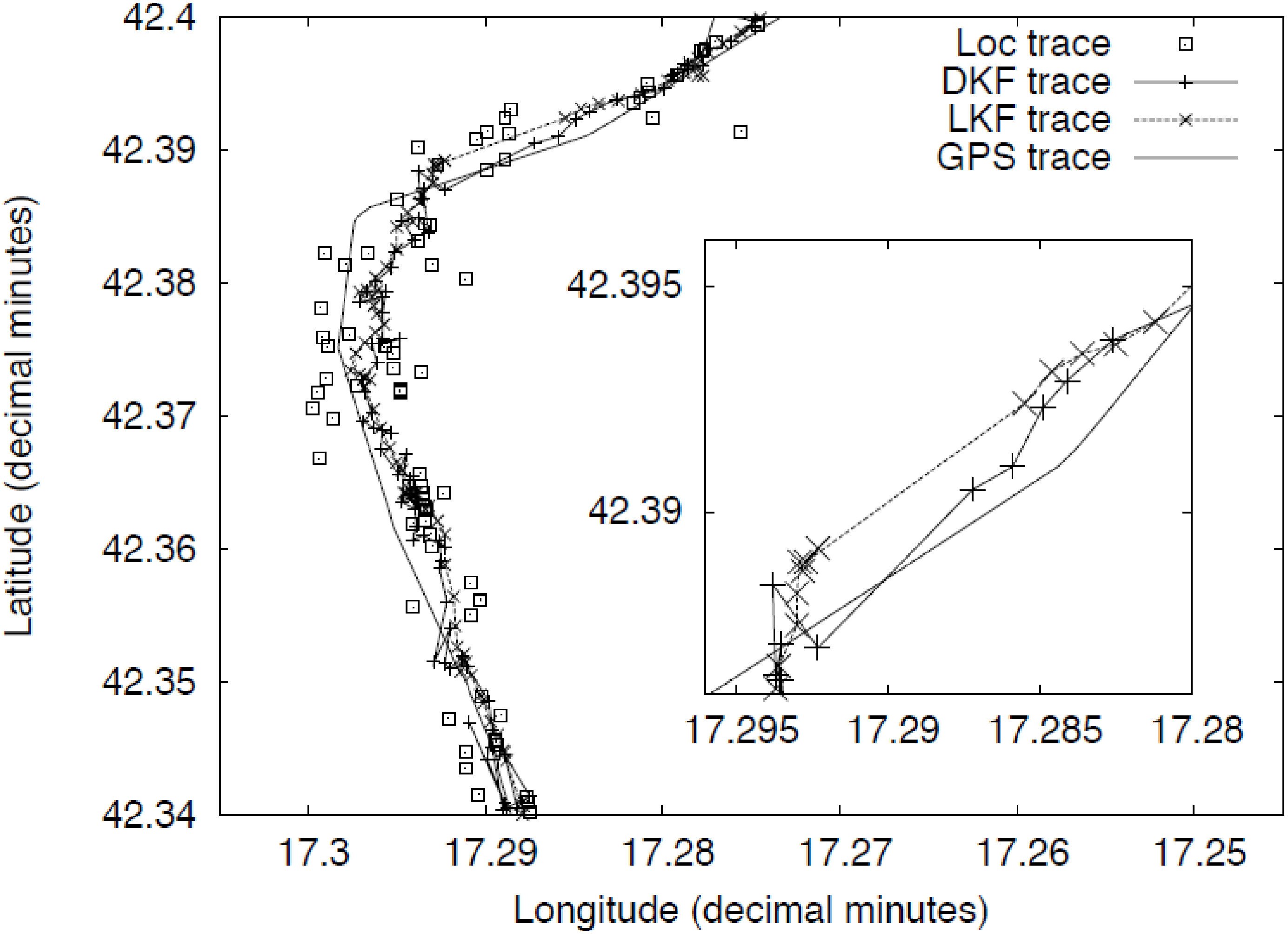
- Two prediction mechanisms
  - Distributed Kalman filter (DKF)
  - Broadcast of the estimates and consensus on an “average” value
  - Local Kalman filter (LKF)
  - Every node calculates it's own prediction

Experiment

Experiment



Evaluation

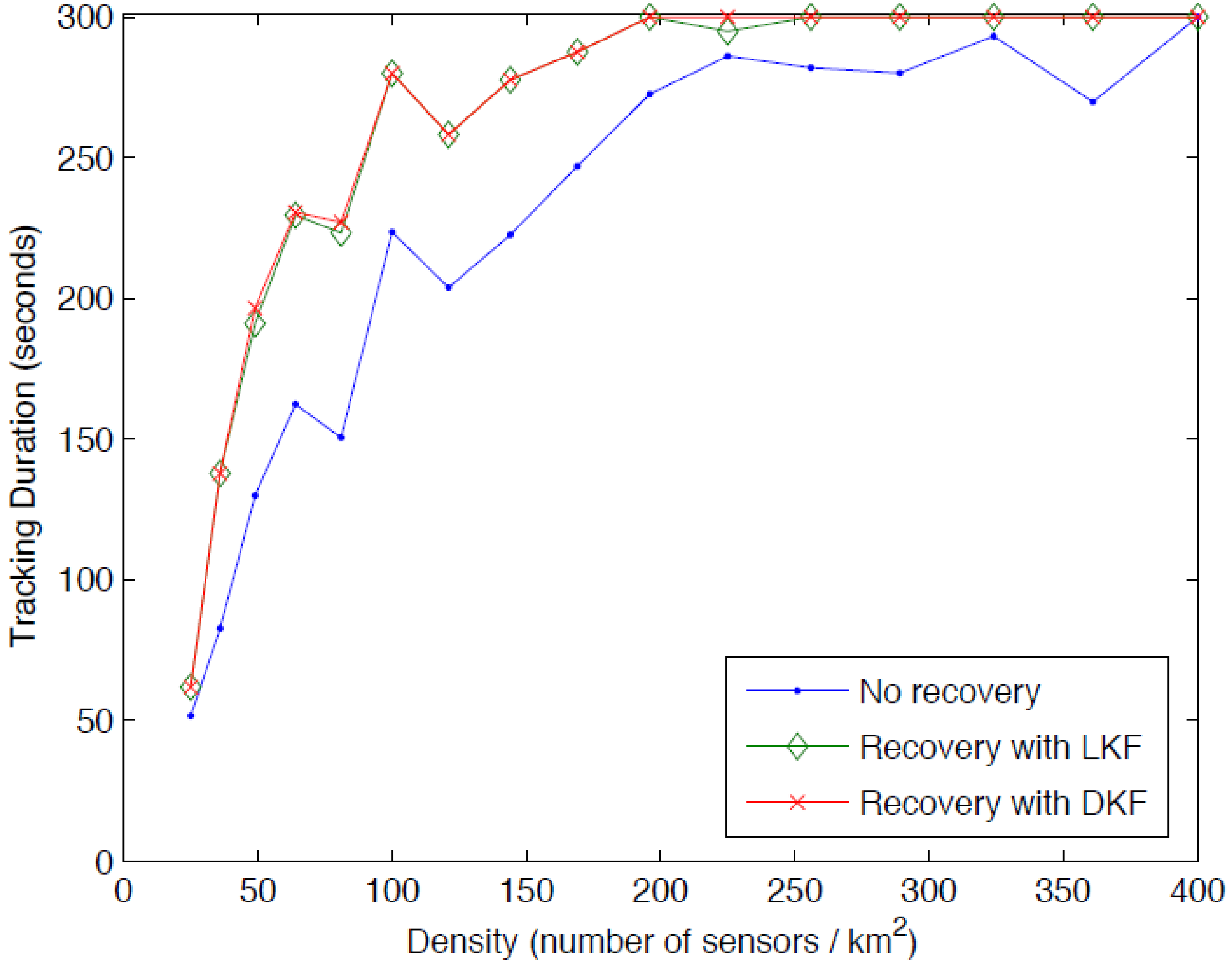


Loc trace
DKF trace
LKF trace
GPS trace
42.4
42.39
42.38
42.37
42.36
42.35
42.34
Latitude (decimal minutes)
17.3
17.29
17.28
17.27
17.26
17.25
Longitude (decimal minutes)
42.395
42.39
17.295
17.29
17.285
17.28

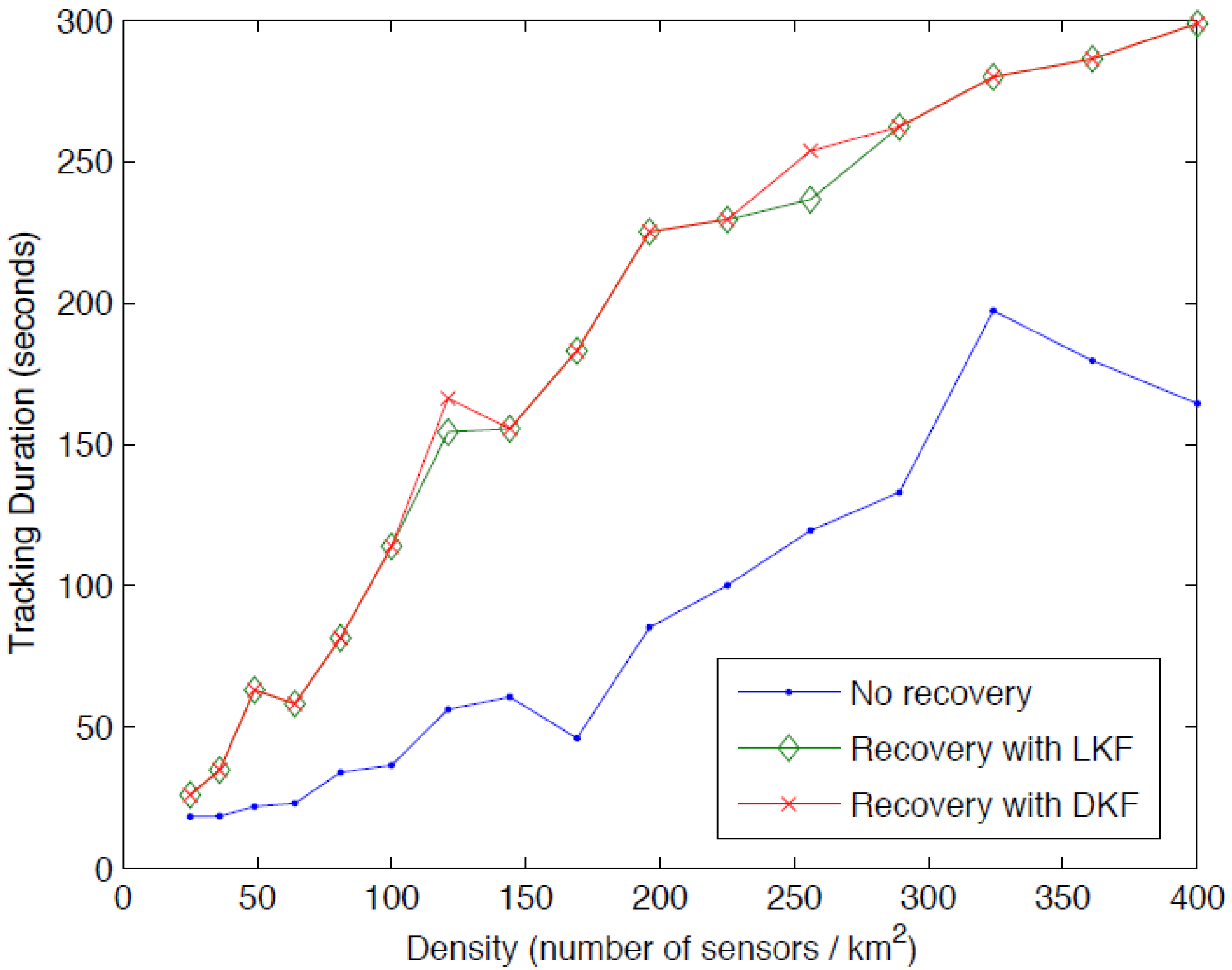
Simulation

- Duration: 300 s = 5 min
- Area: 1 km^2
- Sensing range: 50 and 100 m
- Localization error of event detection
    - Standard deviation: 20 m
- Constant velocity model
- Recovery: w/o, local Kalman filter, and distributed Kalman filter

Tracking duration – 100m sensing range



Tracking duration – 50m sensing range



Simulation outcome

- DKF no advantage over LKF regarding tracking duration
- Limitations:
- 200 – 400 sensors / km^2
  - Zürich: 4239 pop. per km^2 => ~ 9.5 %
  - Sensing range in the real-life test: 20 m
- Equally distributed sensors
- Constant velocity model
  - Authors claim no difference to Manhattan and Random Way-point model

Future Work

- Stated by the authors
  - Incentive for people to opt in
  - Privacy, trust, and security issues
  - GPS calibration
- Further ideas
  - Quantification of energy consumption
  - And optimization
  - Large scale evaluation in real-life environment
  - Improving sensing range and/or needed sensor density

Questions?

Sources

- Picture N95: http://static.trustedreviews.com/94%7cda81b4%7c8cc2_4497-Nokian95lowmenu.jpg
- Population density Zürich: http://de.wikipedia.org/wiki/Z%C3%BCrich
- MetroTrack protocol, experiment, and simulation including diagrams and pictures:
  MetroTrack: Predictive Tracking of Mobile Events using Mobile Phones [Ahn 2010]